

# 面向对象

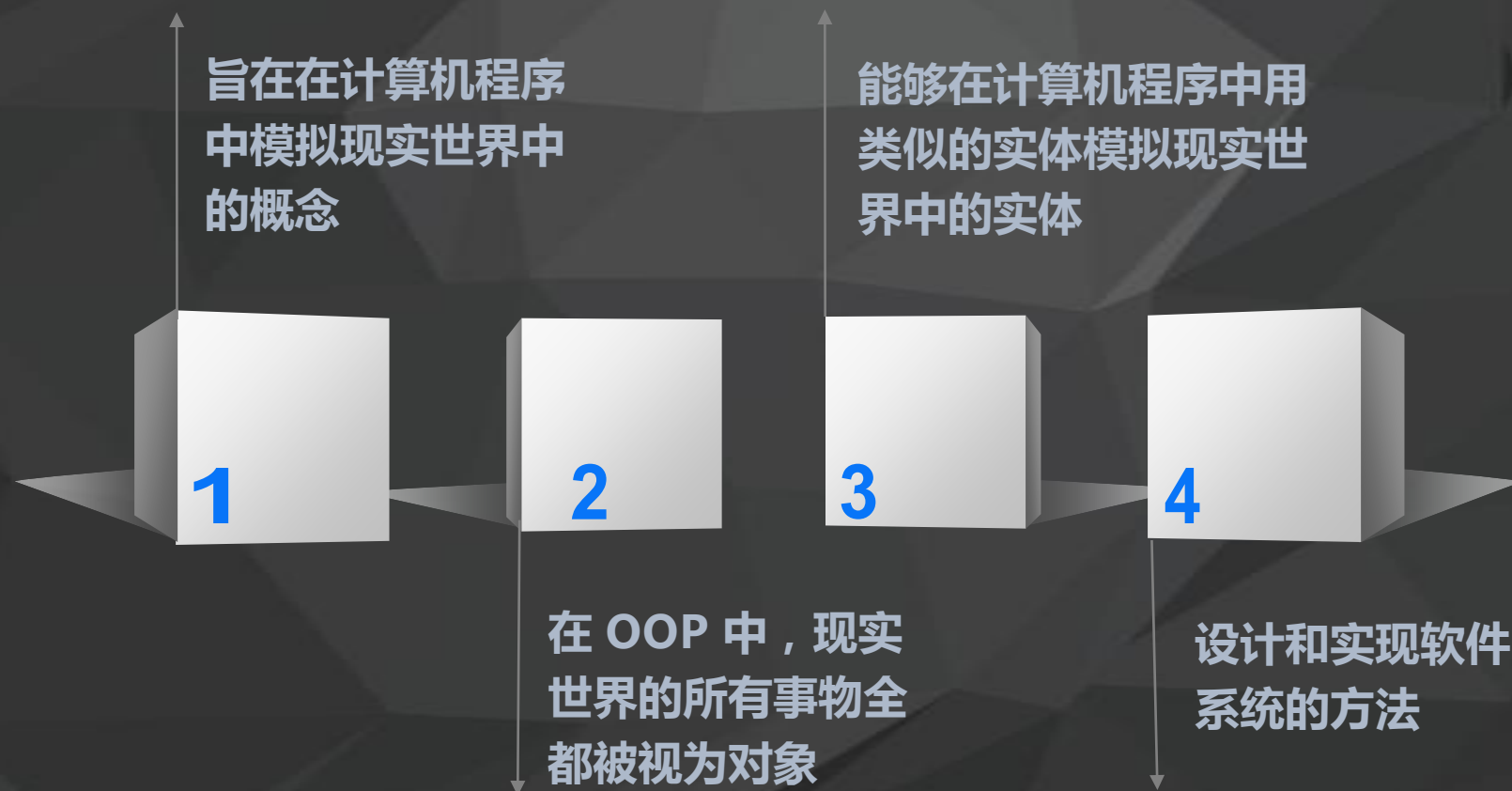


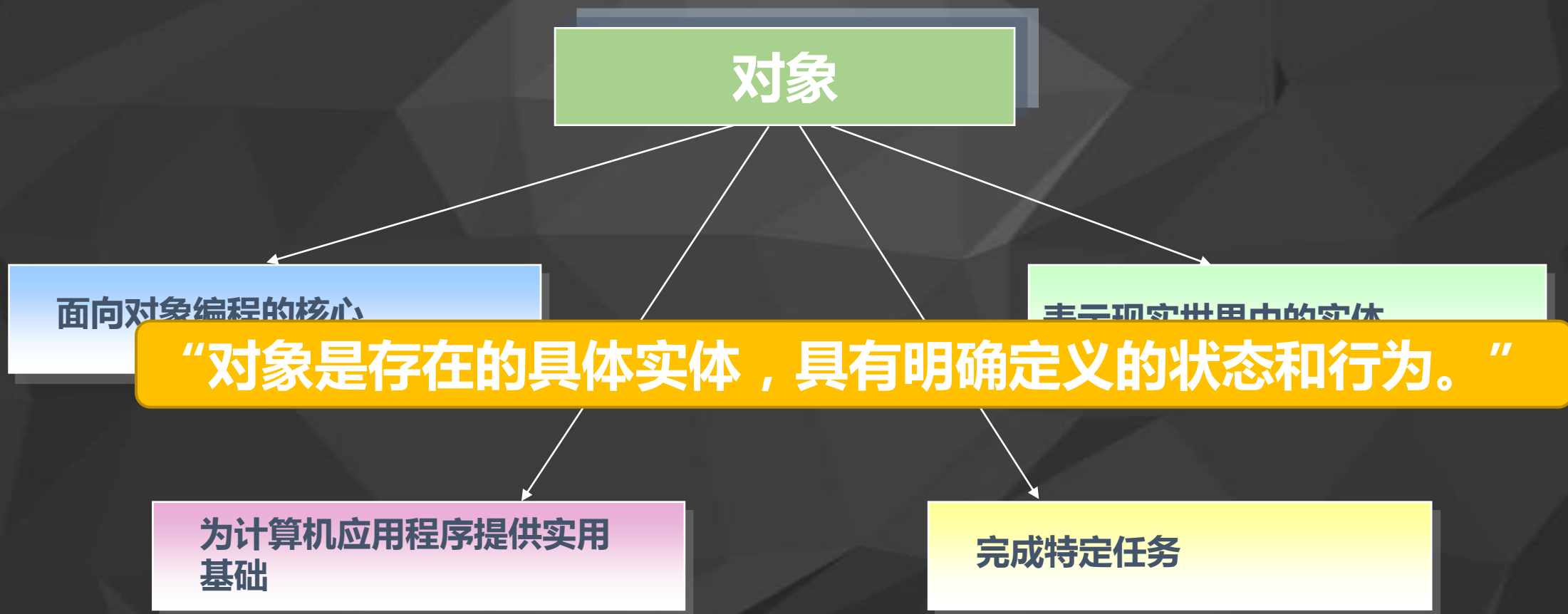
# 回顾

- ▶ Java 中的数据类型分为两个类别：基本数据类型、引用数据类型
- ▶ 变量是存储数据的基本单元
- ▶ 数据转换有两种类型：自动类型转换和强制类型转换
- ▶ 数组用于将相同数据类型的多个数据存储在连续的内存单元中
- ▶ Java提供了一组丰富的运算符
- ▶ Java支持以下控制语句编程结构：
  - ▶ if....else      switch...case
  - ▶ while          do...while          for

- ◆ 理解对象
- ◆ 理解类
- ◆ 理解对象与类之间的关系
- ◆ 初步理解抽象和封装

# 面向对象编程(OOP)





# 对象的示例

张浩

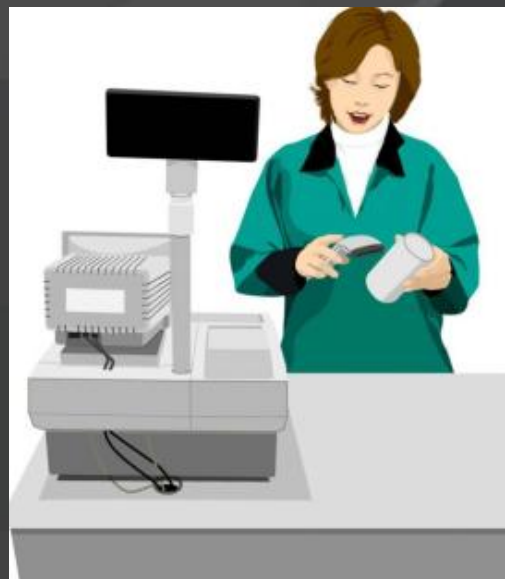


顾客

姓名—张浩  
年龄—20  
体重—60kg

操作：  
购买商品

李明



收银员

员工号—10001  
姓名—李明  
部门—财务部

操作：  
收款  
打印账单

# 对象的特征——属性

- ▶ 属性——对象具有的各种特征
- ▶ 每个对象的每个属性都拥有特定值
  - ▶ 例如：张浩和李明的年龄、姓名不一样



# 对象的特征——方法

## ▶ 方法——对象执行的操作



## ▶ 对象：用来描述客观事物的一个实体，由一组属性和方法构成



# 从对象抽象出“类”

- ▶ 抽取出下列对象的属性和方法的共同特征



轿车





顾客类

轿车类

.....

.....

类是模子，确定对象将会拥  
有的特征（属性）和行为  
（方法）

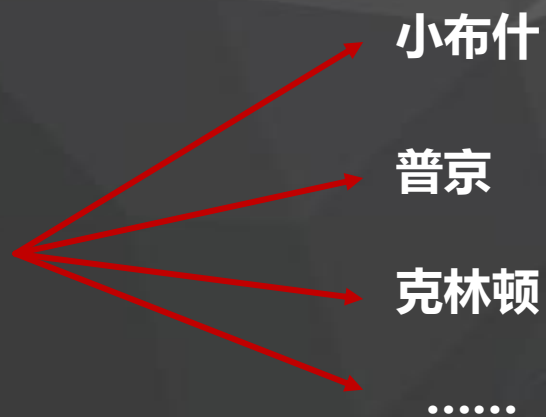
## ▶ 类

- ▶ 具有相同属性和方法的一组对象的集合
- ▶ 类是对象的类型
  - ▶ 不同于int类型：具有方法

# 类和对象的关系

- ▶ 类是抽象的概念，仅仅是模板，比如说：“人”
- ▶ 对象是一个你能够看得到、摸得着的具体实体

“人” 类
特征（属性） 年龄 体重
行为（方法） 衣 食 住 行



## ► 类将现实世界中的概念模拟到计算机程序中

语法：

```
public class 类名 {  
    //定义属性部分  
    属性1的类型 属性1;  
    属性2的类型 属性2;  
    ...  
    属性n的类型 属性n;  
    //定义方法部分  
    方法1;  
    方法2;  
    ...  
    方法m;  
}
```

# 如何定义类

---

## ▶ 定义一个类的步骤

- ▶ 定义类名
- ▶ 编写类的属性
- ▶ 编写类的方法

# ◆ 类 示例

每个学生都有年龄、姓名、班级和爱好。用类的思想编写学生类

**Students类**

name

age

grade

hobby

show()--显示个人信息

# 类 示例

```
public class Students {
```

成员变量

```
String name;    //姓名
```

```
int age;        //年龄
```

```
String grade;   //班级
```

```
String hobby;   //爱好
```

方法

```
//定义学生显示个人信息的方法
```

```
public void show() {
```

```
    System.out.println("我叫：" + name + "，年龄：" + age +  
        "，所在班级：" + grade + "，我的爱好是：" + hobby + "。");
```

定义类的show ()方法，  
用于输出类相关的信息

# ◆ 如何创建和使用对象

## ▶ 使用对象的步骤：

### ▶ 创建对象

类名 对象名 = new 类名();

```
Students zhanghao = new Students ();
```

## ▶ 引用对象成员：使用 “.” 进行以下操作

▶ 引用类的属性：对象名.属性

▶ 引用类的方法：对象名.方法名()

```
zhanghao.name = "张浩"; //给name属性赋值  
zhanghao.show(); //调用show ()方法
```



# 创建和使用对象 示例

## 创建“张浩”对象

```
public class Test {  
    public static void main(String[] args)  
    {  
        Students stu1= new Student();  
        System.out.println("***初始化成员变量前***");  
        stu1.show();  
        stu1.name = "张浩";  
        stu1.age=24;  
        stu1.grade="005";  
        stu1.hobby="写代码";  
        System.out.println("\n***初始化成员变量后***");  
        stu1.show();  
    }  
}
```

说一说看到  
什么效果？

说一说又看到  
什么效果？

- ▶ 类和对象的关系？
- ▶ 如何创建和使用对象？



# 练习

- ▶ 编写学生类，输出学生相关信息；编写教师类，输出教师相关信息

```
控制台
<已终止> InitialStudent [Java 应用程序] C:\Program
张浩
年龄：10
就读于：s1班
爱好：篮球
```

```
控制台
<已终止> InitialTeacher [Java 应用程序] C:\Program
王老师
专业方向：计算机
教授课程：使用Java语言理解程序逻辑
教龄：5
```

## 学生类

姓名 String  
年龄 int  
班级 String  
爱好 String

显示学员个人信息

## 教员类

姓名 String  
专业方向 String  
教授课程 String  
教龄 int

显示教员个人信息

# ◆ 面向对象（OO）的优点

- ▶ **与人类的思维习惯一致**
  - ▶ 把人类解决问题的思维过程转变为程序能够理解的过程
- ▶ **信息隐藏，提高了程序的可维护性和安全性**
  - ▶ 封装实现了模块化和信息隐藏
  - ▶ 封装使得在对象外部不能随意访问对象的属性和方法
- ▶ **提高了程序的可重用性**
  - ▶ 一个类可以创建多个对象实例，增加了重用性

# 数据类型总结

int  
double  
char  
boolean

基本数据类型

String  
数组  
使用class定义的类型  
.....

引用数据类型



# 数据类型使用区别

数据类型	特 征	声 明	使 用
基本数据类型	没有方法	<code>int i= 8;</code>	<code>i++ ;</code>
引用数据类型	具有方法	<code>Student stu = new Student();</code>	<code>stu.age=20;</code>



# 总结

---

- ▶ 什么是类？
- ▶ 什么是对象？
- ▶ 类和对象的区别是什么？
- ▶ 如何定义一个类、类的属性和方法？
- ▶ 如何创建类的对象？
- ▶ 数据类型有哪两种？区别是什么？

# 谢 谢

